Fig.1.

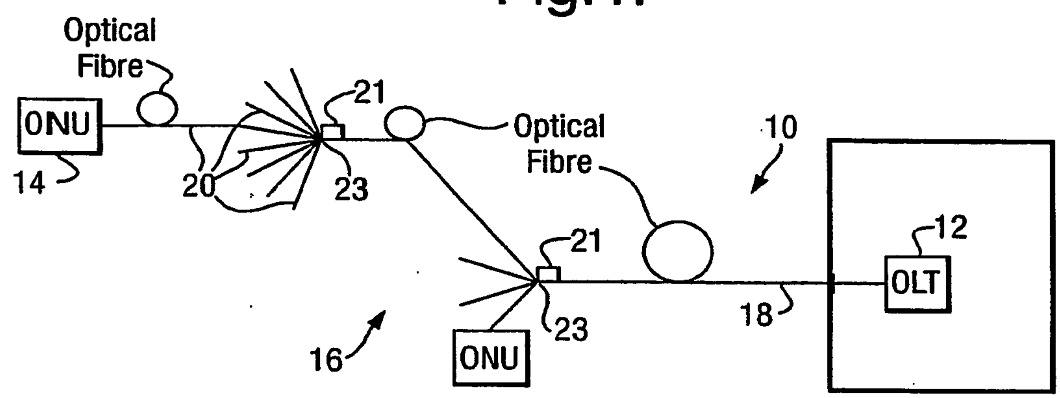


Fig.2a.

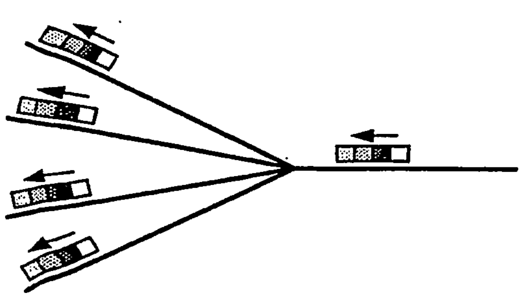


Fig.2b.

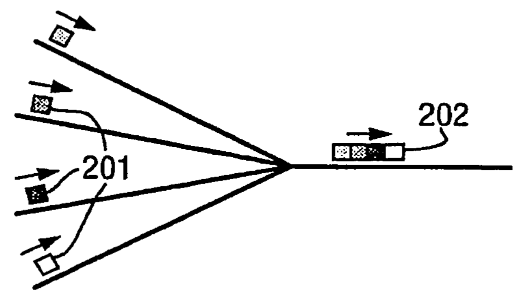


Fig.3a.

PLOAM cells contain grants, message fields, sync bytes and CRC check bytes

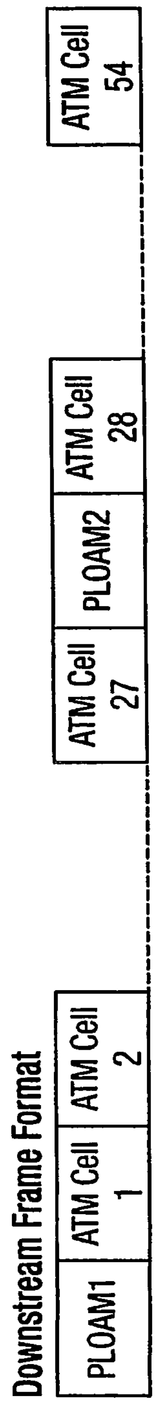


Fig.3b.

Overhead, preamble, delimiting bits and a "guard band" between cells designated "□"

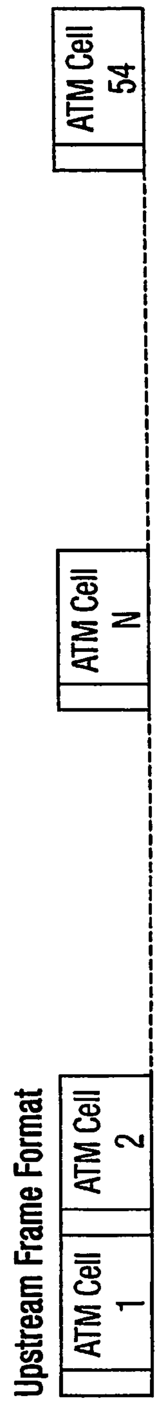


Fig.4.

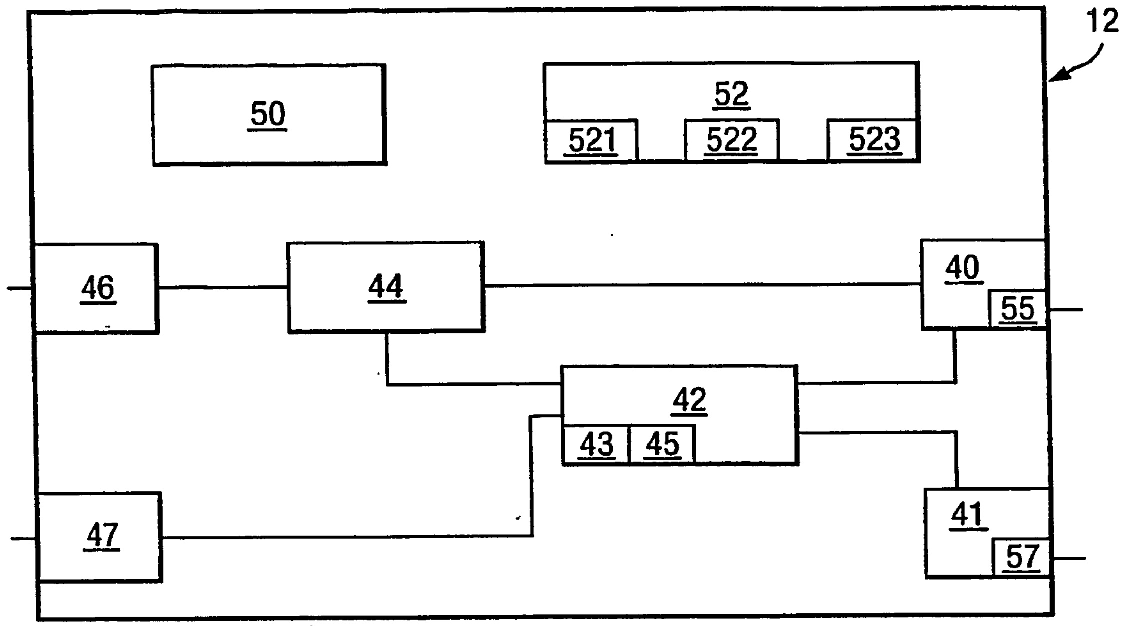

12
50
52
521
522
523
46
44
40
55
42
43
45
47
41
57

Fig.5.

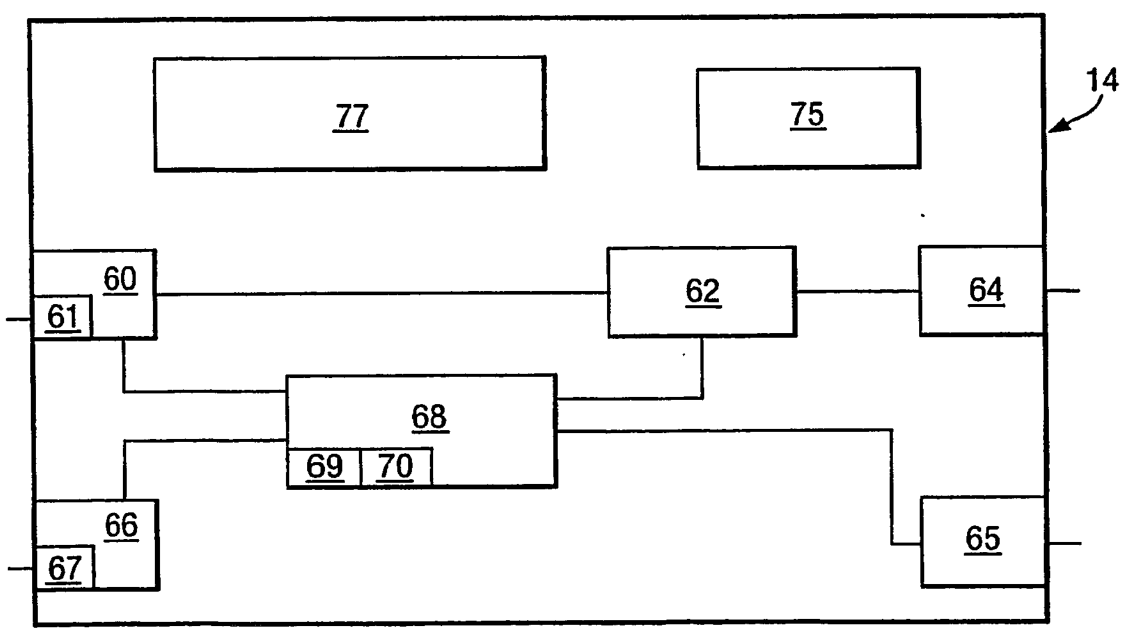

14
77
75
60
61
62
64
68
69
70
66
67
65

Fig.6.

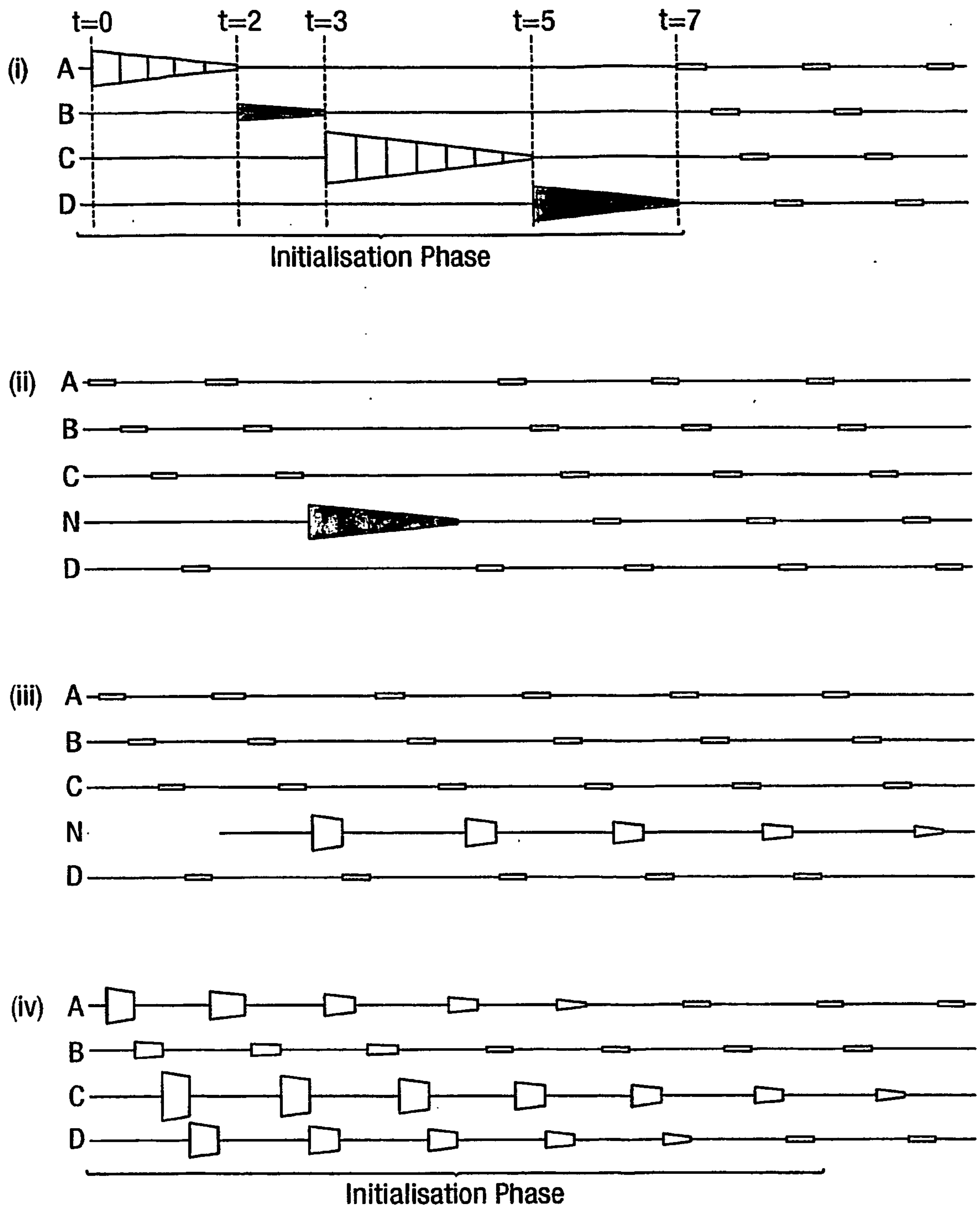

t=0
t=2
t=3
t=5
t=7
(i)
A
B
C
D
Initialisation Phase
(ii)
A
B
C
N
D
(iii)
A
B
C
N
D
(iv)
A
B
C
D
Initialisation Phase

Fig.7.

STEP	Description	M1	M2	M3	c_0	c_1	c_2	$E(t_x) = c_0 t_x + c_1 t_{x-1} + c_2 t_{x-2}$
	Initially all memory locations are set to zero. Coefficients fixed.	0	0	0	c_0	c_1	c_2	
1a	Shift Data (redundant this time)	0	0	0	c_0	c_1	c_2	
1b	Sample signal at position x, save to first data location	R1	0	0	c_0	c_1	c_2	
1c	Calculate equalised data, E	R1	0	0	c_0	c_1	c_2	c_0.R1
2a	Shift data	0	R1	0	c_0	c_1	c_2	
2b	Sample signal at position x+1, save to first data location	R2	R1	0	c_0	c_1	c_2	
2c	Calculate equalised data, E	R2	R1	0	c_0	c_1	c_2	c_0.R2 + c_1.R1
3a	Shift data	0	R2	R1	c_0	c_1	c_2	
3b	Sample signal at position x+2, save to first data location	R3	R2	R1	c_0	c_1	c_2	
3c	Calculate equalised data, E	R3	R2	R1	c_0	c_1	c_2	c_0.R3 + c_1.R2 + c_2.R1
4	Shift data, Sample signal at position x+3, save to first data location, Calculate equalised data, E	R4	R3	R2	c_0	c_1	c_2	c_0.R4 + c_1.R3 + c_2.R2
5	Shift data, Sample signal at position x+4, save to first data location, Calculate equalised data, E	R5	R4	R3	c_0	c_1	c_2	c_0.R5 + c_1.R4 + c_2.R3
6	Shift data, Sample signal at position x+5, save to first data location, Calculate equalised data, E	R6	R5	R4	c_0	c_1	c_2	c_0.R6 + c_1.R5 + c_2.R4
7	Shift data, Sample signal at position x+6, save to first data location, Calculate equalised data, E	R7	R6	R5	c_0	c_1	c_2	c_0.R7 + c_1.R6 + c_2.R5

Fig.8a.

STEP	Description	M1	M2	M3	c_0	c_1	c_2
0	Initially all memory locations are set to zero. Coefficients set to their initial values	0	0	0	$c_0(0)$	$c_1(0)$	$c_2(0)$
1a	Shift Data (redundant this time)	0	0	0	$c_0(0)$	$c_1(0)$	$c_2(0)$
1b	Sample signal at position x, save to first data location	R1	0	0	$c_0(0)$	$c_1(0)$	$c_2(0)$
1c	Calculate equalised data, E	R1	0	0	$c_0(0)$	$c_1(0)$	$c_2(0)$
1d	Calculate "error", e	R1	0	0	$c_0(0)$	$c_1(0)$	$c_2(0)$
1e	Calculate gradient, δc	R1	0	0	$c_0(0)$	$c_1(0)$	$c_2(0)$
1f	Tweak Coefficients $c=c+\delta c$	R1	0	0	$c_0(1) = c_0(0) + \delta c_0$	$c_1(1) = c_1(0) + \delta c_1$	$c_2(1) = c_2(0) + \delta c_2$
2a	Shift Data	0	R1	0	$c_0(1)$	$c_1(1)$	$c_2(1)$
2b	Sample signal at position x+1, save to first data location	R2	R1	0	$c_0(1)$	$c_1(1)$	$c_2(1)$
2c	Calculate equalised data, E	R2	R1	0	$c_0(1)$	$c_1(1)$	$c_2(1)$
2d	Calculate "error", e	R2	R1	0	$c_0(1)$	$c_1(1)$	$c_2(1)$
2e	Calculate gradient, δc	R2	R1	0	$c_0(1)$	$c_1(1)$	$c_2(1)$
2f	Tweak Coefficients $c=c+\delta c$	R2	R1	0	$c_0(2) = c_0(1) + \delta c_0$	$c_1(2) = c_1(1) + \delta c_1$	$c_2(2) = c_2(1) + \delta c_2$
3a	Shift Data	0	R2	R1	$c_0(2)$	$c_1(2)$	$c_2(2)$
3b	Sample signal at position x+2, save to first data location	R3	R2	R1	$c_0(2)$	$c_1(2)$	$c_2(2)$
3c	Calculate equalised data, E	R3	R2	R1	$c_0(2)$	$c_1(2)$	$c_2(2)$
3d	Calculate "error", e	R3	R2	R1	$c_0(2)$	$c_1(2)$	$c_2(2)$
3e	Calculate gradient, δc	R3	R2	R1	$c_0(2)$	$c_1(2)$	$c_2(2)$
3f	Tweak Coefficients $c=c+\delta c$	R3	R2	R1	$c_0(3) = c_0(2) + \delta c_0$	$c_1(3) = c_1(2) + \delta c_1$	$c_2(3) = c_2(2) + \delta c_2$
4a	Shift Data	0	R3	R2	$c_0(3)$	$c_1(3)$	$c_2(3)$
4b	Sample signal at position x+3, save to first data location	R4	R3	R2	$c_0(3)$	$c_1(3)$	$c_2(3)$
4c	Calculate equalised data, E	R4	R3	R2	$c_0(3)$	$c_1(3)$	$c_2(3)$
4d	Calculate "error", e	R4	R3	R2	$c_0(3)$	$c_1(3)$	$c_2(3)$
4e	Calculate gradient, δc	R4	R3	R2	$c_0(3)$	$c_1(3)$	$C_2(3)$
4f	Tweak Coefficients $c=c+\delta c$	R4	R3	R2	$c_0(4) = c_0(3) + \delta c_0$	$c_1(4) = c_1(3) + \delta c_1$	$c_2(4) = c_2(3) + \delta c_2$

Fig.8b.

STEP	Description	Error, e	Gradient	E
0	Initially all memory locations are set to zero. Coefficients set to their initial values			
1a	Shift Data (redundant this time)			
1b	Sample signal at position x, save to first data location			
1c	Calculate equalised data, E			$E(1) = c_0(0).R1$
1d	Calculate "error", e	$e(1) = K(1) - E(1)$		
1e	Calculate gradient, δc		$\delta c = (\delta c_0, \delta c_1, \delta c_2)$	
1f	Tweak Coefficients $c=c+\delta c$			
2a	Shift Data			
2b	Sample signal at position x+1, save to first data location			
2c	Calculate equalised data, E			$E(2) = c_0(1).R2 + c_1(1).R1$
2d	Calculate "error", e	$e(2) = K(2) - E(2)$		
2e	Calculate gradient, δc		$\delta c = (\delta c_0, \delta c_1, \delta c_2)$	
2f	Tweak Coefficients $c=c+\delta c$			
3a	Shift Data			
3b	Sample signal at position x+2, save to first data location			
3c	Calculate equalised data, E			$E(3) = c_0(2).R3 + c_1(2).R2 + c_2(2).R1$
3d	Calculate "error", e	$e(3) = K(3) - E(3)$		
3e	Calculate gradient, δc		$\delta c = (\delta c_0, \delta c_1, \delta c_2)$	
3f	Tweak Coefficients $c=c+\delta c$			
4a	Shift Data			
4b	Sample signal at position x+3, save to first data location			
4c	Calculate equalised data, E			$E(4) = c_0(3).R4 + c_1(3).R3 + c_2(3).R2$
4d	Calculate "error", e	$e(4) = K(4) - E(4)$		
4e	Calculate gradient, δc		$\delta c = (\delta c_0, \delta c_1, \delta c_2)$	
4f	Tweak Coefficients $c=c+\delta c$			

Fig.9a.

Step		Description	M1	M2	M3	ONU
	II	Determine ONU				j
	III	Retrieve coefficients for ONU				j
i	IV	Shift Data	0	0	0	J
	V	Sample signal, save to first data location	Ri	0	0	J
	VI	Calculate equalised data, E	Ri	0	0	J
	VII	Calculate "error", e	Ri	0	0	J
	VIII	Calculate gradient, δc	Ri	0	0	J
	IX	Tweak Coefficients c=c+δc	Ri	0	0	J
	X	End of Cell? [No]	Ri	0	0	J
i+1	IV	Shift Data	0	Ri	0	J
	V	Sample signal, save to first data location	Ri+1	Ri	0	J
	VI	Calculate equalised data, E	Ri+1	Ri	0	j
	VII	Calculate "error", e	Ri+1	Ri	0	j
	VIII	Calculate gradient, δc	Ri+1	Ri	0	j
	IX	Tweak Coefficients c=c+δc	Ri+1	Ri	0	j
	X	End of Cell? [No]	Ri+1	Ri	0	j
i+2	IV	Shift Data	0	Ri+1	Ri	j
	V	Sample signal, save to first data location	Ri+2	Ri+1	Ri	j
	VI	Calculate equalised data, E	Ri+2	Ri+1	Ri	j
	VII	Calculate "error", e	Ri+2	Ri+1	Ri	j
	VIII	Calculate gradient, δc	Ri+2	Ri+1	Ri	j
	IX	Tweak Coefficients c=c+δc	Ri+2	Ri+1	Ri	j
	X	End of Cell? [No]	Ri+2	Ri+1	Ri	j
i+3	IV	Shift Data	0	Ri+2	Ri+1	j
	V	Sample signal, save to first data location	Ri+3	Ri+2	Ri+1	j
	VI	Calculate equalised data, E	Ri+3	Ri+2	Ri+1	j
	VII	Calculate "error", e	Ri+3	Ri+2	Ri+1	j
	VIII	Calculate gradient, δc	Ri+3	Ri+2	Ri+1	j
	IX	Tweak Coefficients c=c+δc	Ri+3	Ri+2	Ri+1	j
	X	End of Cell? [No]	Ri+3	Ri+2	Ri+1	j
		Etc, etc, etc, ...	"	"	"	j
	X	End of Cell? [Yes]	"	"	"	j
		Store coefficients for ONU j				j
		Start loop for new cell				

Fig.9a (Cont).

Step		Description	M1	M2	M3	ONU
M	II	Determine ONU				n
M	III	Retrieve coefficients for ONU				n
M	IV	Shift Data	0	0	0	n
	V	Sample signal, save to first data location	Rm	0	0	n
	VI	Calculate equalised data, E	Rm	0	0	n
	VII	Calculate "error", e	Rm	0	0	n
	VIII	Calculate gradient, δc	Rm	0	0	n
	IX	Tweak Coefficients c=c+δc	Rm	0	0	n
	X	End of Cell? [No]	Rm	0	0	n
M+ 1	IV	Shift Data	0	Rm	0	n
	V	Sample signal, save to first data location	Rm+1	Rm	0	n
	VI	Calculate equalised data, E	Rm+1	Rm	0	n
	VII	Calculate "error", e	Rm+1	Rm	0	n
	VIII	Calculate gradient, δc	Rm+1	Rm	0	n
	IX	Tweak Coefficients c=c+δc	Rm+1	Rm	0	n
	X	End of Cell? [No]	Rm+1	Rm	0	n
M+ 2	IV	Shift Data Etc, etc, etc	0	Rm+1	Rm	n

Fig.9b.

Step		c_0	c_1	c_2	Error	Gradient	E
	II						
	III	$c_0(j,i)$	$c_1(j,i)$	$c_2(j,i)$			
i	IV	$c_0(j,i)$	$c_1(j,i)$	$c_2(j,i)$			
	V	$c_0(j,i)$	$c_1(j,i)$	$c_2(j,i)$			
	VI	$c_0(j,i)$	$c_1(j,i)$	$c_2(j,i)$			E(i)
	VII	$c_0(j,i)$	$c_1(j,i)$	$c_2(j,i)$	c		
	VIII	$c_0(j,i)$	$c_1(j,i)$	$c_2(j,i)$		δc	
	IX	$c_0(j,i+1) = c_0(j,i) + \delta c_0$	$c_1(j,i+1) = c_1(j,i) + \delta c_1$	$c_2(j,i+1) = c_2(j,i) + \delta c_2$			
	X	$c_0(j,i+1)$	$c_1(j,i+1)$	$c_2(j,i+1)$			
i+1	IV	$c_0(j,i+1)$	$c_1(j,i+1)$	$c_2(j,i+1)$			
	V	$c_0(j,i+1)$	$c_1(j,i+1)$	$c_2(j,i+1)$			
	VI	$c_0(j,i+1)$	$c_1(j,i+1)$	$c_2(j,i+1)$			E(i+1)
	VII	$c_0(j,i+1)$	$c_1(j,i+1)$	$c_2(j,i+1)$	e		
	VIII	$c_0(j,i+1)$	$c_1(j,i+1)$	$c_2(j,i+1)$		δc	
	IX	$c_0(j,i+2) = c_0(j,i+1) + \delta c_0$	$c_1(j,i+2) = c_1(j,i+1) + \delta c_1$	$c_2(j,i+2) = c_2(j,i+1) + \delta c_2$			
	X	$c_0(j,i+2)$	$c_1(j,i+2)$	$c_2(j,i+2)$			
i+2	IV	$c_0(j,i+2)$	$c_1(j,i+2)$	$c_2(j,i+2)$			
	V	$c_0(j,i+2)$	$c_1(j,i+2)$	$c_2(j,i+2)$			
	VI	$c_0(j,i+2)$	$c_1(j,i+2)$	$c_2(j,i+2)$			E(i+2)
	VII	$c_0(j,i+2)$	$c_1(j,i+2)$	$c_2(j,i+2)$	e		
	VIII	$c_0(j,i+2)$	$c_1(j,i+2)$	$c_2(j,i+2)$		δc	
	IX	$c_0(j,i+3) = c_0(j,i+2) + \delta c_0$	$c_1(j,i+3) = c_1(j,i+2) + \delta c_1$	$c_2(j,i+3) = c_2(j,i+2) + \delta c_2$			
	X	$c_0(j,i+3)$	$c_1(j,i+3)$	$c_2(j,i+3)$			
i+3	IV	$c_0(j,i+3)$	$c_1(j,i+3)$	$c_2(j,i+3)$			
	V	$c_0(j,i+3)$	$c_1(j,i+3)$	$c_2(j,i+3)$			
	VI	$c_0(j,i+3)$	$c_1(j,i+3)$	$c_2(j,i+3)$			E(i+3)
	VII	$c_0(j,i+3)$	$c_1(j,i+3)$	$c_2(j,i+3)$	e		
	VIII	$c_0(j,i+3)$	$c_1(j,i+3)$	$c_2(j,i+3)$		δc	
	IX	$c_0(j,i+4) = c_0(j,i+3) + \delta c_0$	$c_1(j,i+4) = c_1(j,i+3) + \delta c_1$	$c_2(j,i+4) = c_2(j,i+3) + \delta c_2$			
	X	$c_0(j,i+4)$	$c_1(j,i+4)$	$c_2(j,i+4)$			
	X	“ “	“ “	“ “			

Fig.9b (Cont).

Step		c_0	c_1	c_2	Error	Gradient	E
		Store coefficients for ONU j					
M M	II III	$c_0(n,m)$	$c_1(n,m)$	$c_2(n,m)$			
M	IV	$c_0(n,m)$	$c_1(n,m)$	$c_2(n,m)$			
	V	$c_0(n,m)$	$c_1(n,m)$	$c_2(n,m)$			
	VI	$c_0(n,m)$	$c_1(n,m)$	$c_2(n,m)$			E(m)
	VII	$c_0(n,m)$	$c_1(n,m)$	$c_2(n,m)$	e		
	VIII	$c_0(n,m)$	$c_1(n,m)$	$c_2(n,m)$		δc	
	IX	$c_0(n,m+1) = c_0(n,m) + \delta c_0$	$c_1(n,m+1) = c_1(n,m) + \delta c_1$	$c_2(n,m+1) = c_2(n,m) + \delta c_2$			
	X	$c_0(n,m+1)$	$c_1(n,m+1)$	$c_2(n,m+1)$			
M+1	IV	$c_0(n,m+1)$	$c_1(n,m+1)$	$c_2(n,m+1)$			
	V	$c_0(n,m+1)$	$c_1(n,m+1)$	$c_2(n,m+1)$			
	VI	$c_0(n,m+1)$	$c_1(n,m+1)$	$c_2(n,m+1)$			E(m+1)
	VII	$c_0(n,m+1)$	$c_1(n,m+1)$	$c_2(n,m+1)$	e		
	VIII	$c_0(n,m+1)$	$c_1(n,m+1)$	$c_2(n,m+1)$		δc	
	IX	$c_0(n,m+2) = c_0(n,m+1) + \delta c_0$	$c_1(n,m+2) = c_1(n,m+1) + \delta c_1$	$c_2(n,m+2) = c_2(n,m+1) + \delta c_2$			
	X	$c_0(n,m+2)$	$c_1(n,m+2)$	$c_2(n,m+2)$			
M+2	IV	$c_0(n,m+2)$	$c_1(n,m+2)$	$c_2(n,m+2)$			

Fig.10.

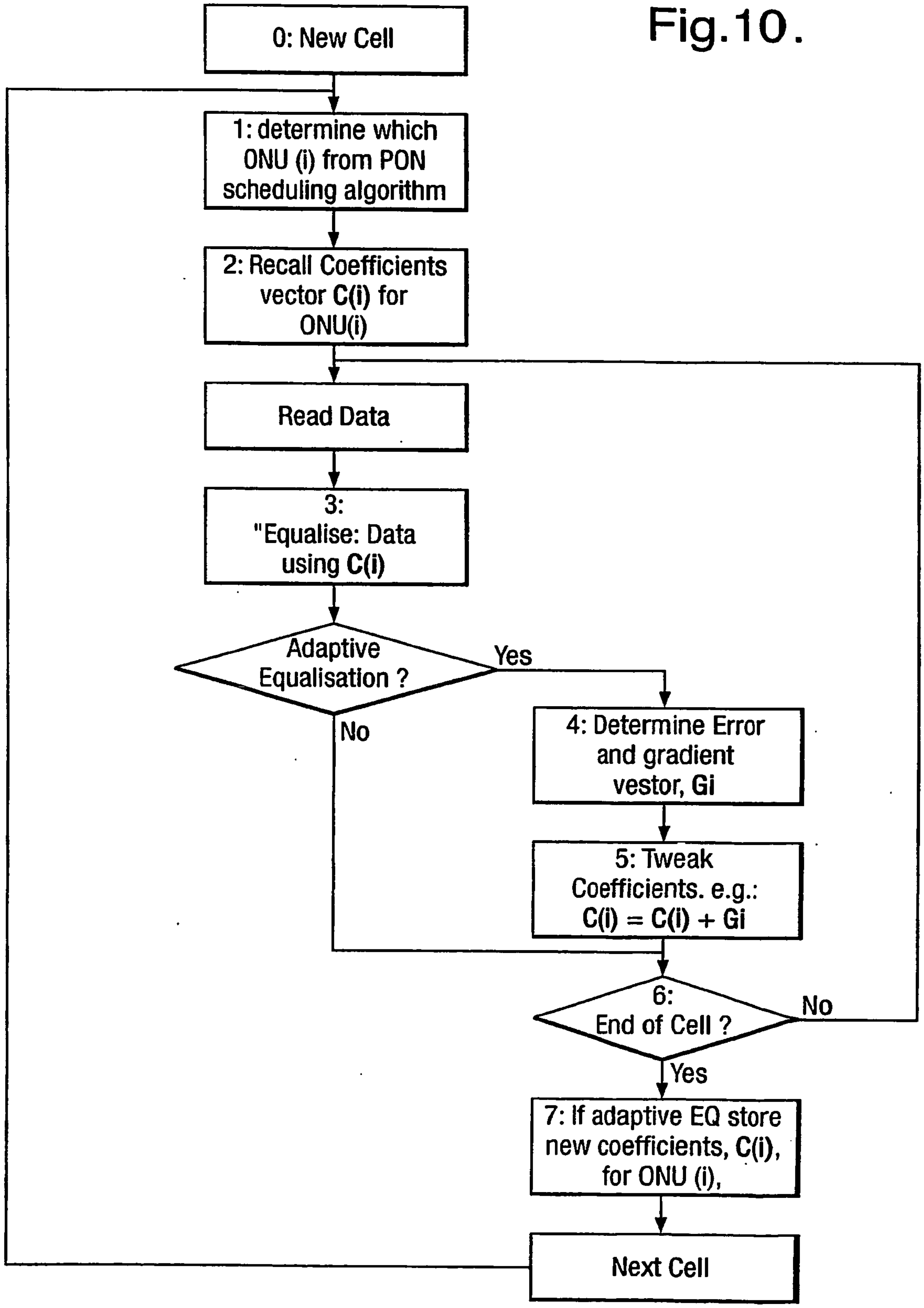

0: New Cell
1: determine which ONU (i) from PON scheduling algorithm
2: Recall Coefficients vector C(i) for ONU(i)
Read Data
3: "Equalise: Data using C(i)
Adaptive Equalisation ?
Yes
No
4: Determine Error and gradient vestor, Gi
5: Tweak Coefficients. e.g.: C(i) = C(i) + Gi
6: End of Cell ?
No
Yes
7: If adaptive EQ store new coefficients, C(i), for ONU (i),
Next Cell

Fig.11.

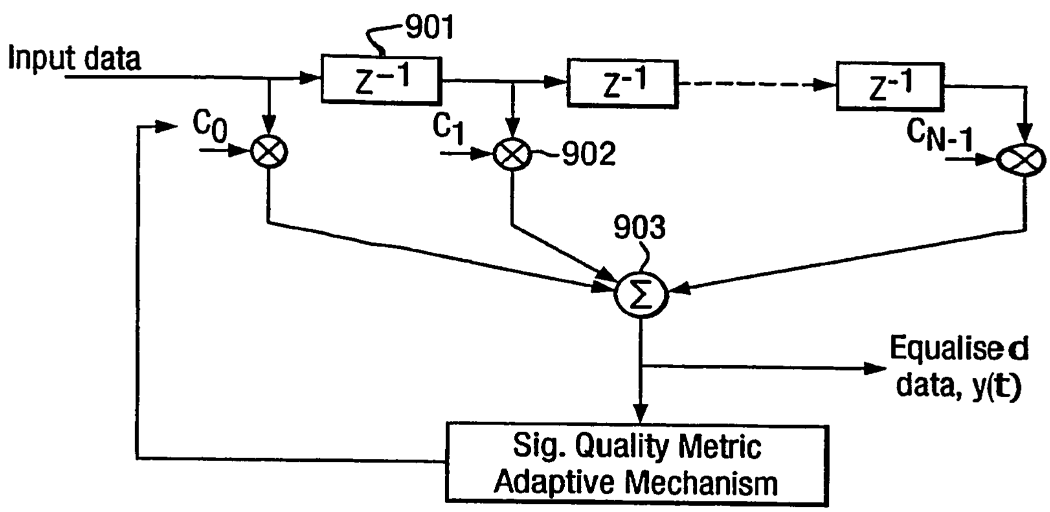


Fig.12a.

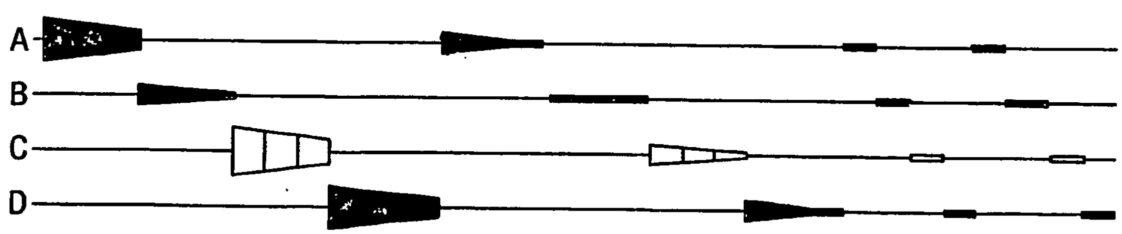


Fig.12b.

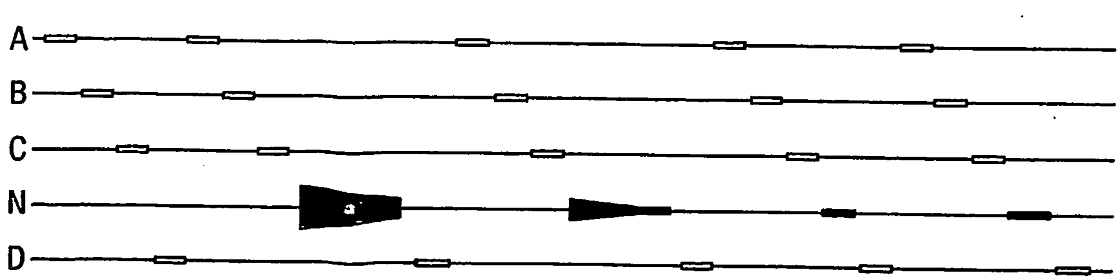


Fig.13.

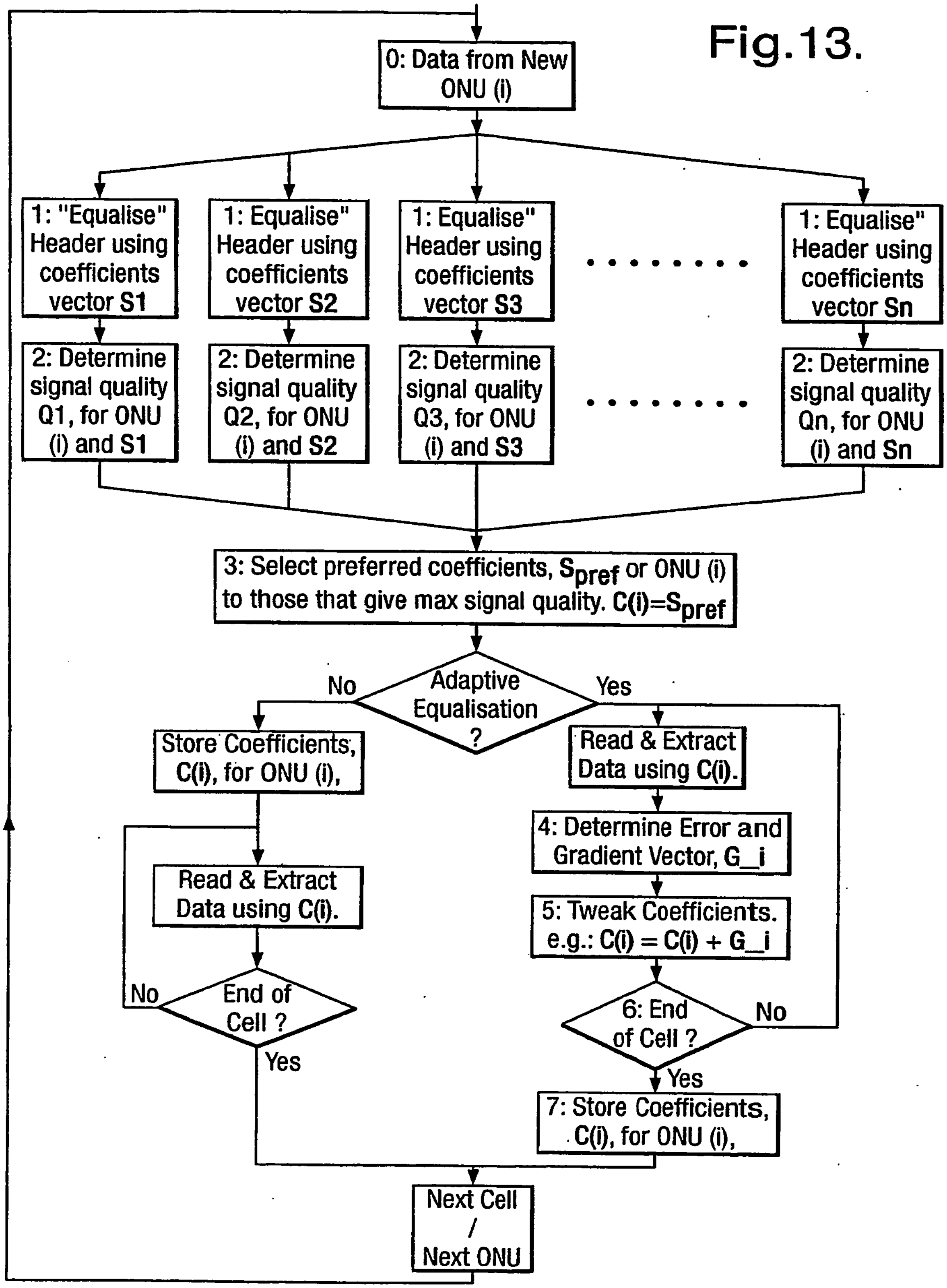
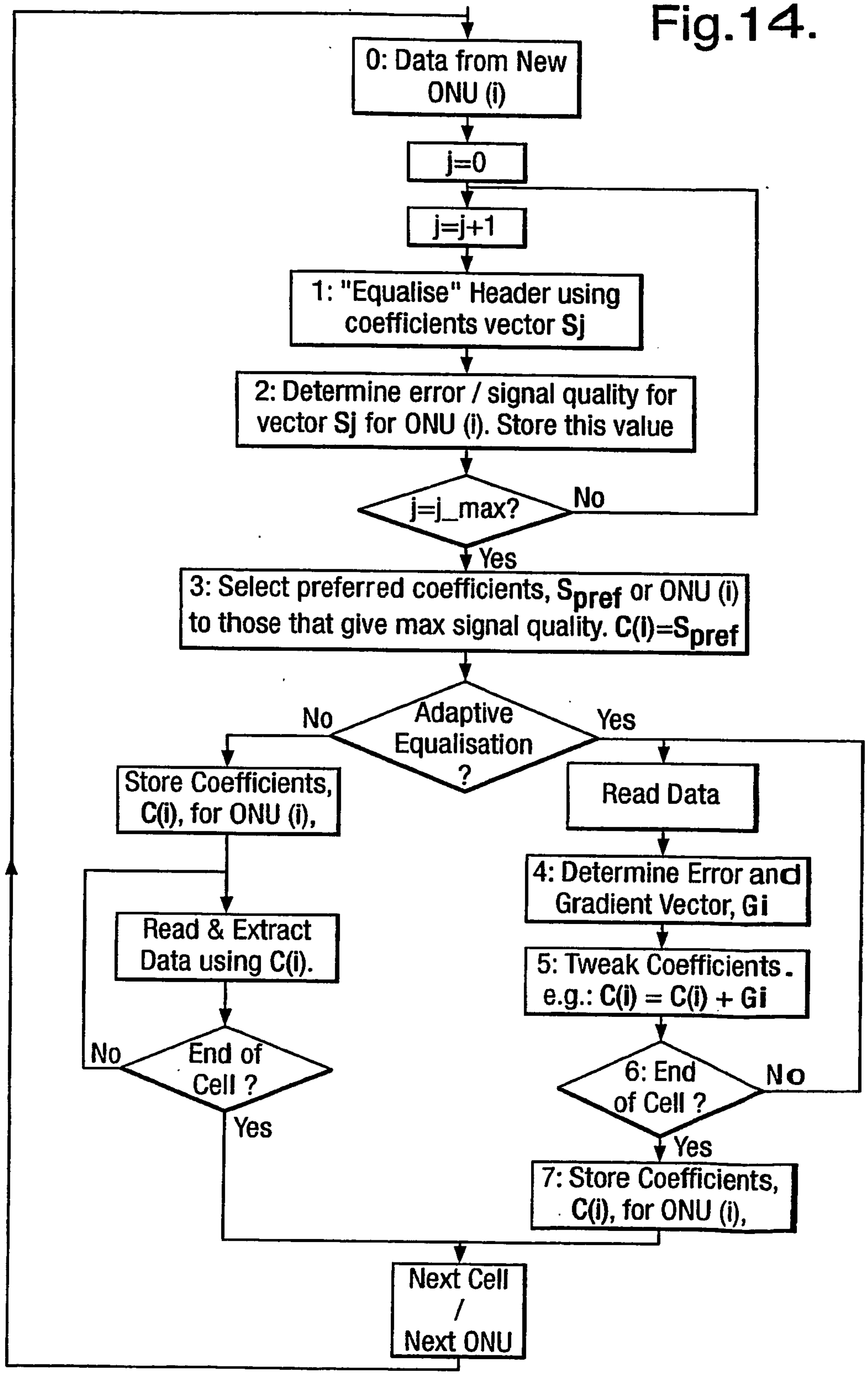

0: Data from New ONU (i)
1: "Equalise" Header using coefficients vector S1
1: Equalise" Header using coefficients vector S2
1: Equalise" Header using coefficients vector S3
1: Equalise" Header using coefficients vector Sn
2: Determine signal quality Q1, for ONU (i) and S1
2: Determine signal quality Q2, for ONU (i) and S2
2: Determine signal quality Q3, for ONU (i) and S3
2: Determine signal quality Qn, for ONU (i) and Sn
3: Select preferred coefficients, Spref or ONU (i) to those that give max signal quality. C(i)=Spref
Adaptive Equalisation ?
No
Yes
Store Coefficients, C(i), for ONU (i),
Read & Extract Data using C(i).
End of Cell ?
No
Yes
Read & Extract Data using C(i).
4: Determine Error and Gradient Vector, G_i
5: Tweak Coefficients. e.g.: C(i) = C(i) + G_i
6: End of Cell ?
No
Yes
7: Store Coefficients, C(i), for ONU (i),
Next Cell / Next ONU

Fig.14.


0: Data from New ONU (i)
j=0
j=j+1
1: "Equalise" Header using coefficients vector Sj
2: Determine error / signal quality for vector Sj for ONU (i). Store this value
j=j_max?
No
Yes
3: Select preferred coefficients, Spref or ONU (i) to those that give max signal quality. C(I)=Spref
Adaptive Equalisation ?
No
Yes
Store Coefficients, C(i), for ONU (i),
Read Data
4: Determine Error and Gradient Vector, Gi
Read & Extract Data using C(i).
5: Tweak Coefficients. e.g.: C(i) = C(i) + Gi
End of Cell ?
No
Yes
6: End of Cell ?
No
Yes
7: Store Coefficients, C(i), for ONU (i),
Next Cell / Next ONU

16/16

Fig.15(i).

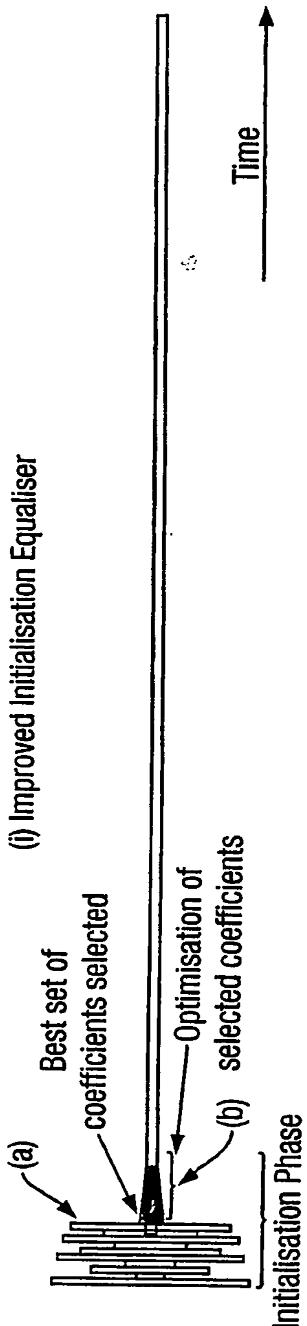


Fig.15(ii).